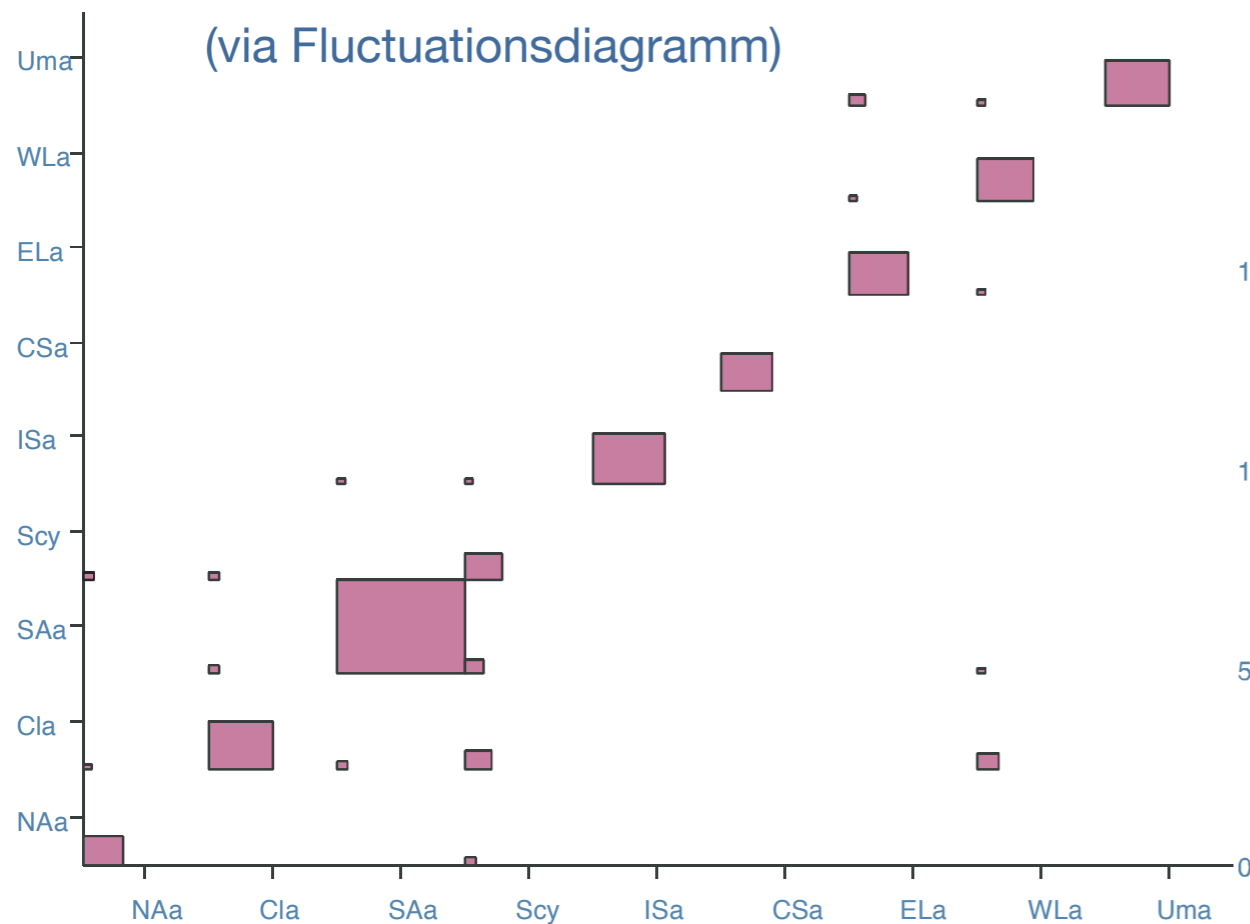


CART: Diagnose

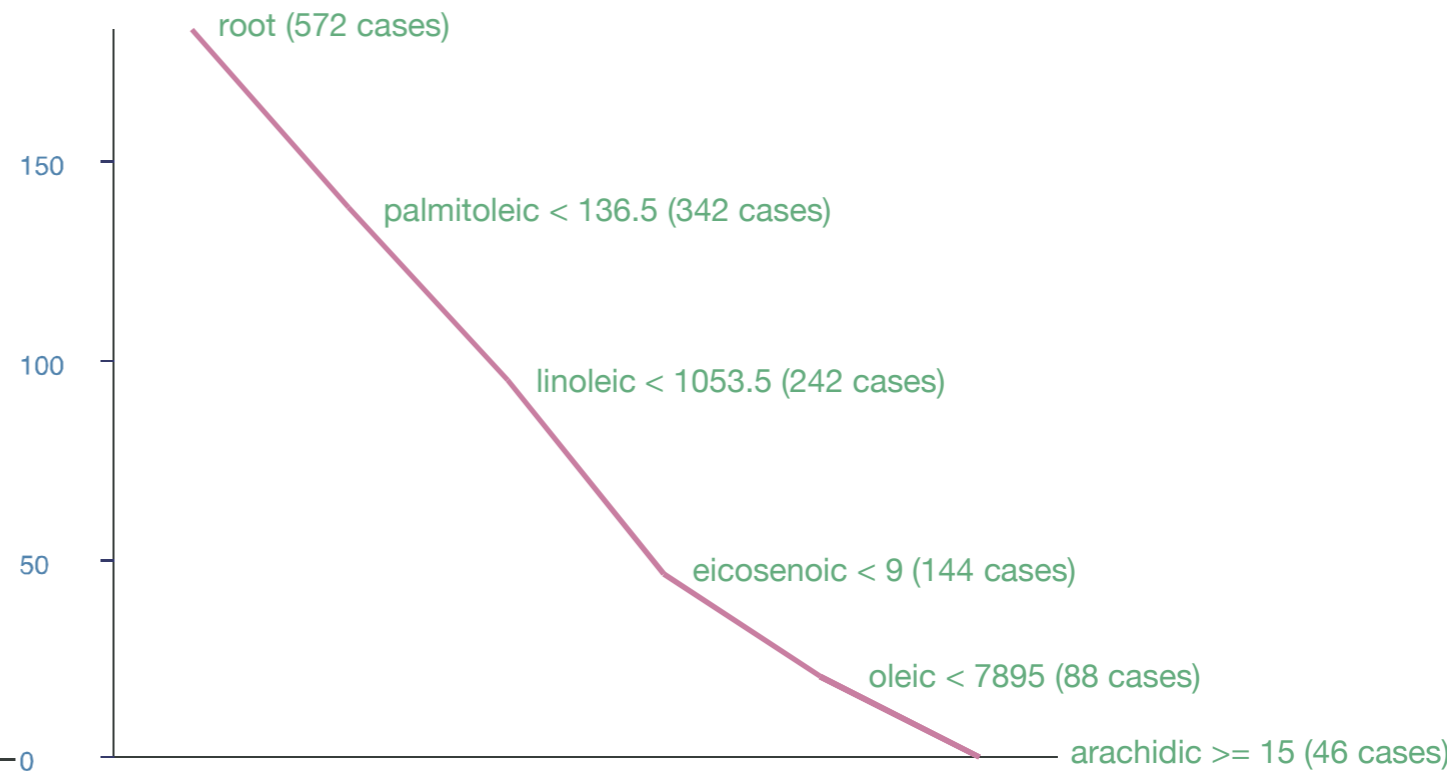
- Konfusionsmatrix ✓
Für *Region* war der Baum auf S.217 fehlerfrei.
- Sectioned Scatterplot ✓
- Neues Problem: Vorhersage von *Area*.

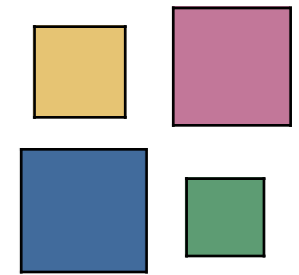
Konfusionsmatrix

(via Fluctuationsdiagramm)



Deviance Plot für einen Knoten

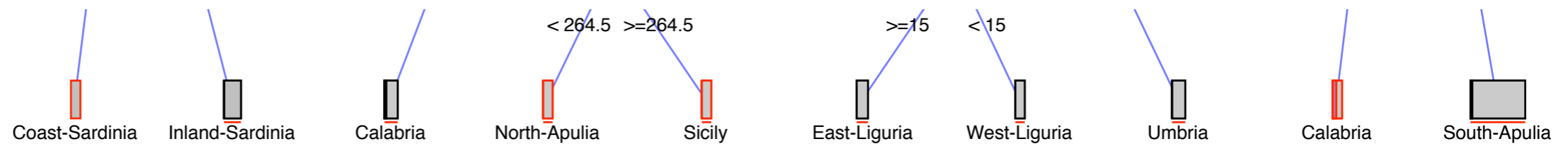




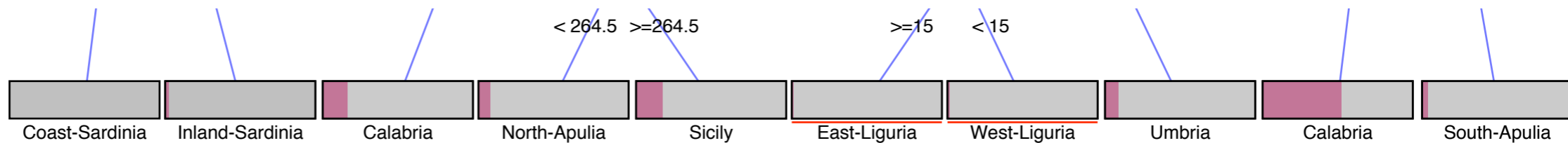
CART: Diagnose cont.

Rest Devianz des Baumes summiert sich über die Blätter (alle falsch klassifizierte Fälle selektiert!)

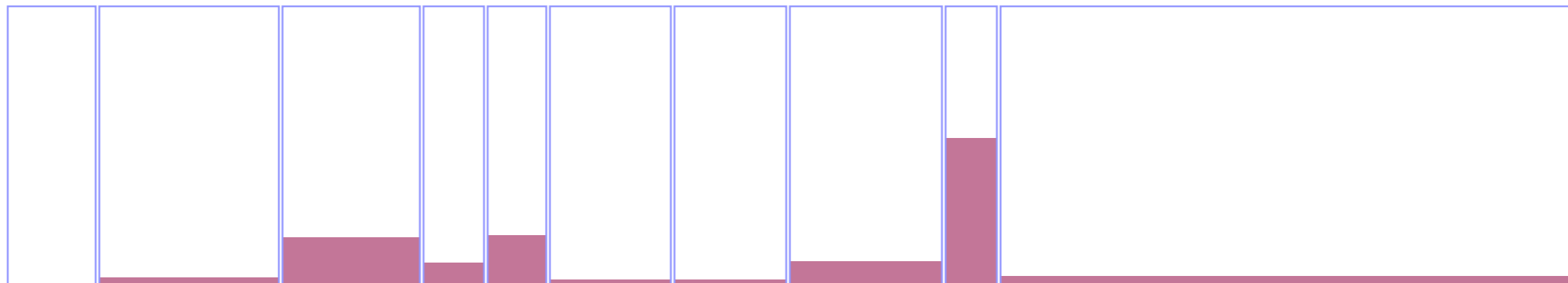
Fallzahl proportionale Blätter

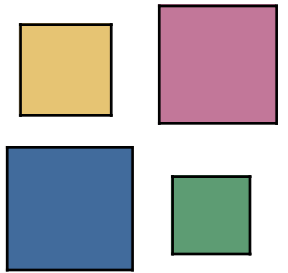


Fixe Größe der Blätter



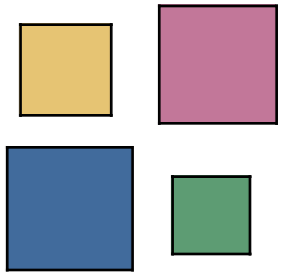
Spineplot der Blätter





Pruning

- Als Pruning (kürzen, beschneiden, reduzieren) bezeichnet man den Vorgang, in dem Bäume verkleinert werden, d.h. Splits werden wieder entfernt.
- Zu kleiner Devianz-Gewinn wird im allgemeinen nicht als Stopp Kriterium verwendet, da ein folgender Split evtl. wieder einen großen Devianz Gewinn haben könnte.
 - ↳ erzeuge erst einen “zu großen” Baum, der später wieder zurückgeschnitten (prune) wird.
- Pruning ist auch von der grundsätzlichen Instabilität eines Baums betroffen.
- Lösung: Cost Complexity Pruning mit Cross-Validation



Cross Validation

Der gesamte Datensatz wird in K Gruppen geteilt. Das Modell wird schrittweise für $k = 1, \dots, K$ angepasst, jeweils die k -te Gruppe herausgenommen, der Fehler aber auf Gruppe k bewertet.

$\hat{f}^{-k}(x_i)$ sei das geschätzte Modell ohne die Werte aus Gruppe k .

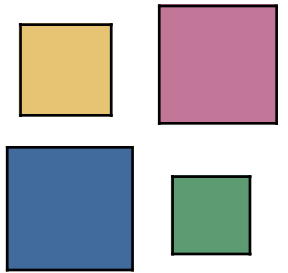
Sei $\kappa : \{1, \dots, N\} \rightarrow \{1, \dots, k\}$ die Indexfunktion, die jede Beobachtung ihrer Gruppe zuordnet, und $L(y_i, \hat{f}^{-k}(x_i))$ eine Verlustfunktion, die den Fehler zwischen y_i und der Schätzung $\hat{f}^{-\kappa(i)}(x_i)$ misst.

Dann ist die Cross-Validation (CV) $CV = \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{f}^{-\kappa(i)}(x_i))$ Sei

α nun ein komplexitäts Parameter der Modellklasse aus der \hat{f} stammt, so wird mittels

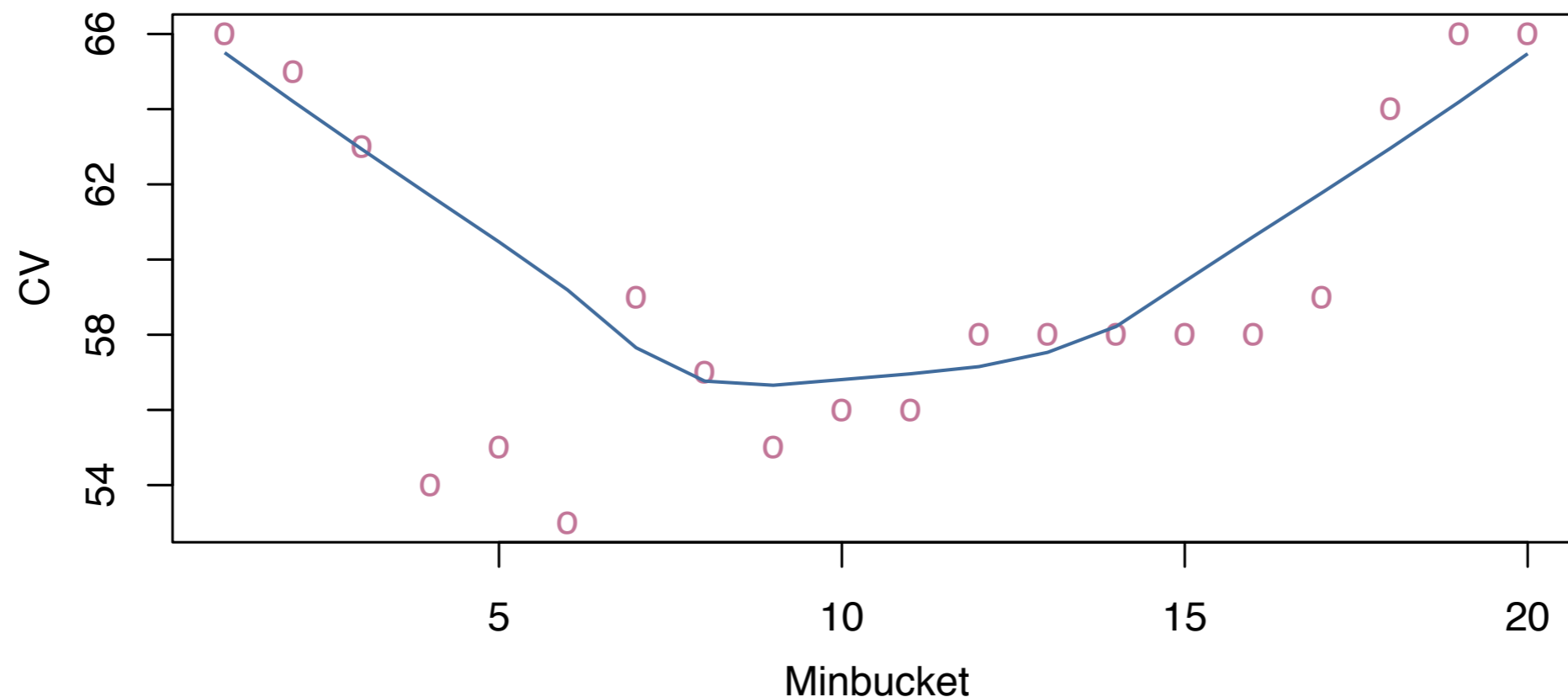
$$CV(\alpha) = \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{f}^{-\kappa(i)}(x_i, \alpha))$$

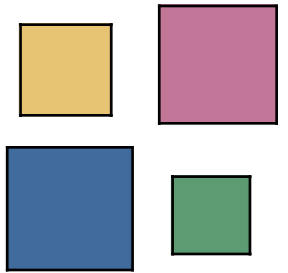
α so gewählt, dass $CV(\alpha)$ minimiert wird.



Cross Validation: Olive Oils

- Teile den Datensatz in 15 Teile \Rightarrow 15-fold cross validation
- Berechne den Baum auf 14/15 der Daten, teste ihn auf 1/15.
- Fehler: Summe aller falsch klassifizierten Daten.
- Komplexitätsparameter α : Minimale Anzahl der Beobachtungen in einem Blatt.





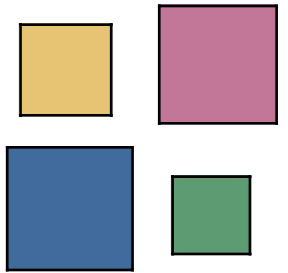
Cross Validation: Olive Oils - R Code

```
n<-length(olives[,1])
fold <- 15
chunk <- trunc(n/fold)
nn <- chunk*fold
runs <- 20

Area <- as.factor(Area)

CV <- matrix(0, runs, 1)

for(i in 1:runs) {
  for(j in 1:fold) {
    test <- (1:chunk)*fold-fold+j
    train <- (1:nn)[-test]
    rpt <- rpart(Area ~ palmitic + palmitoleic + stearic +
                 oleic + linoleic + linolenic +
                 arachidic + eicosenoic, subset=train,
                 cp=0.000001, minsplit=i, minbucket=i)
    conf <- as.matrix(table(predict(rpt,
                                   newdata=olives[test,4:11],
                                   type="class"),
                          Area[test]))
    CV[i] <- CV[i] + sum(conf - diag(diag(conf), 9,9))
  }
}
plot(CV, xlab="Minbucket", pch="o")
lines(lowess(CV ~ (1:runs)))
```



Cost Complexity Pruning

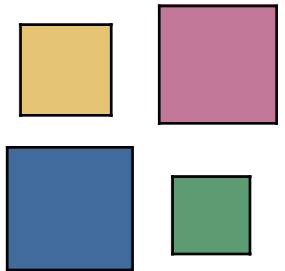
Sei T ein beliebiger Baum, der durch Pruning des Baums T_0 entsteht. Bezeichne R_m die Region des Blattes m , und $|T|$ die Anzahl der Blätter des Baumes. Dann definiert sich das Cost-Complexity Kriterium als

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|.$$

Zu gegebenem α wird ein Baum $T_\alpha \subseteq T$ gesucht, der $C_\alpha(T)$ minimiert. α verhandelt dabei zwischen optimaler Anpassung durch den Baum und minimaler Baumgröße.

Wird beim Pruning jeweils der “schwächste” Knoten kollabiert (bis zur Wurzel), so kann gezeigt werden, daß auf der Sequenz aller Bäume von T_0 bis zur Wurzel T_α enthalten ist.

$\hat{\alpha}$ wird i.a. durch 5, oder 10 fache Kreuz Validierung bestimmt.



Bäume in R: `rpart` Library

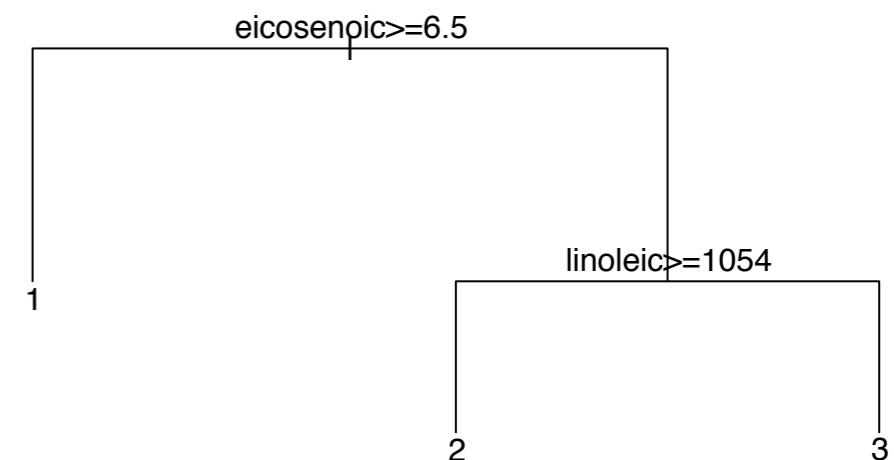
- Die direkt in R verfügbare Funktion `tree` ist fehlerhaft!
- `rpart` arbeitet mittels der in R üblichen “Formelsprache”, und folgt auch sonst den üblichen Modell Objekten.
- `rpart` benutzt den Gini-Index als Unreinheits Maß.
- Olive Oils Beispiel:

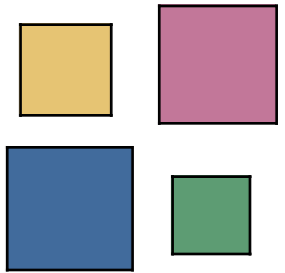
```
> opart <- rpart(Region ~ palmitic + palmitoleic + stearic + oleic +
                linoleic + linolenic + arachidic + eicosenoic)
n= 572
```

```
node), split, n, loss, yval, (yprob)
* denotes terminal node
```

```
1) root 572 249 1 (0.5646853 0.17 0.26)
 2) eicosenoic >= 6.5 323 0 1 (1.0 0.0 0.0) *
 3) eicosenoic < 6.5 249 98 3 (0.0 0.39 0.6)
   6) linoleic >= 1053.5 98 0 2 (0.0 1.0 0.0) *
   7) linoleic < 1053.5 151 0 3 (0.0 0.0 1.0) *
```

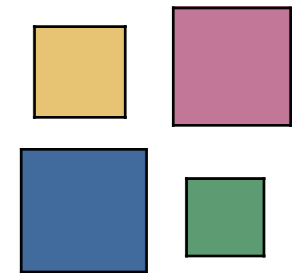
```
> plot(opart)
> text(opart)
```





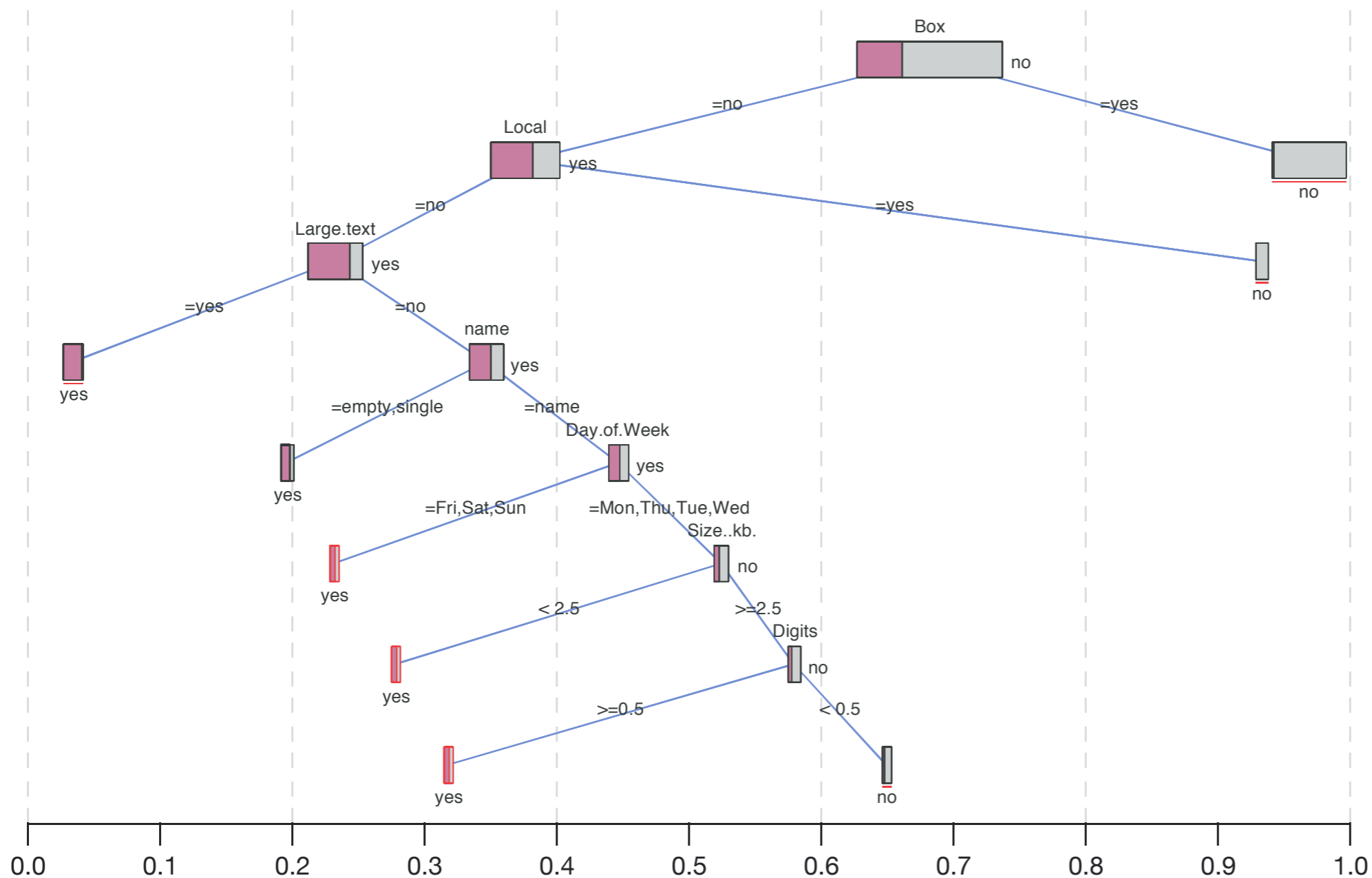
Case Study: Spam e-mails

- 2020 e-mails von 19 Teilnehmern des Statistikkurs 503X in Ames, IA
- Variablen:
 - ~~IS~~id - Eindeutige Id des Kursteilnehmers
 - ~~id~~Eindeutige Id der e-mail je Kursteilnehmer
 - **Day of Week** - Wochentag an dem die e-mail zugestellt wurde
 - **Time of Day** - Uhrzeit an dem die e-mail zugestellt wurde
 - **Box** - Ist eine e-mail des Absenders bereits in der Mailbox des Benutzers?
 - ~~D~~omain - Domain des Absenders, z.B. .com, .de, net ...
 - **Local** - Ist der Absender in der gleichen Domain (in diesem Fall ...iastate.edu)
 - **Digits** - Anzahl der Ziffern im Absender Name
 - **name** - Absender Name, "name", falls richtiger Name, "single", falls nur Vorname, "empty", falls kein Name
 - **%capital** - Prozent der Großbuchstaben im Betreff
 - **Special** - Anzahl der Sonderzeichen im Betreff
 - **credit** - 'yes', falls eines der Wörter 'mortgage', 'sale', 'approve' oder 'credit' im Betreff steht
 - **sucker** - 'yes', falls eines der Wörter 'earn', 'free', oder 'save' im Betreff steht
 - **porn** - 'yes', falls eines der Wörter 'nude', 'sex', 'enlarge' oder 'improve' im Betreff steht
 - **chain** - 'yes', falls eines der Wörter 'pass', 'forward', oder 'help' im Betreff steht
 - **username** - Ist der Name des Empfängers im Betreff?
 - **Large Text** - Werden Zeichensätze größer als Standard verwendet?
 - ~~Spam~~% - Bewertung des ISU internen Spam filters (100%=Spam, 0%=kein Spam)
 - ~~C~~ategory - Art der e-mail: "COMmercial", "LIST-server", "NEWSletter" oder "ORDinary"
 - ~~Spam~~ - Ist die e-mail Spam?

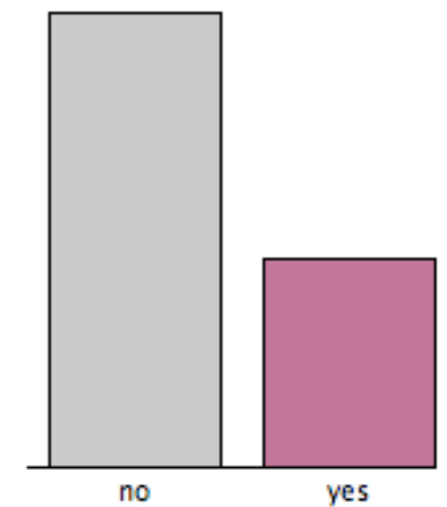


Case Study: Spam e-mails cont.

Knoten und Blätter nach Anteil “non-spam” sortiert.
(Was bedeutet dies für einen potentiellen Spam Filter?)



Spam Mails



Vorhersage

